

REMARKS

Claims 1-4, 7, 8, 10, 12 and 16-19 have been rejected. Claims 5, 6, 9 and 13-15 have been objected to as being dependent upon a rejected base claim. Claims 1-19 are currently pending. Claims 1, 2, 3, 5, 6, 9, 10, 13, 14, and 18 have been amended and claims 20 and 21 have been added.

Specifically, and in accordance with the item numbering in the Office Action, the Office Action has:

In Items 1 and 2, rejected claims 10, 11, 18, and 19 under section 112; Claim 10 is rejected as failing to recited an antecedent basis for “the stored data”; Claim 11 is rejected as it depends on claim 10; claim 18 is rejected because the recitation “at least endpoint associated” is alleged to be unclear; and claim 19 is rejected as it depends on claim 18;

In Items 3 and 4, rejected claims 1-4, 7, 8, 10, 12, and 16-18 under 35 USC §102(e) as being anticipated by Larky ‘495 (U.S. Patent No. 6,389,495); and

In Item 6, objected to claims 5, 6, 9 and 13-15, as being dependent on a rejected base claim, and stated that these claims would be allowable if rewritten in independent form including limitations of the base and any intervening claims.

Items 1 and 2

Applicant thanks the Examiner for pointing out the informalities in the claims. Claim 10 has been amended to provide antecedent basis for the phrase “the stored data.” Claim 18 has been amended, by adding the word “one” to clear up the cited phrase. Applicant believes that amended claims 10 and 18, and dependent claims 11 and 19 are now in good form for allowance.

Items 3 and 4

The Office Action has alleged that each and every limitation of claim 1 is found in the Larky ‘495 reference. The Larky reference describes a dedicated circuit that allows a host computer to configure a software driven USB device. According to FIGs. 7a and 7b of Larky, and the description in the reference relating to these figures, the USB dedicated circuit copies the SETUP data directly to the device RAM buffer 700 in the peripheral device. The device CPU decodes the “get descriptor” request and sets its memory pointer to the descriptor table

702 in RAM. The USB dedicated circuit then completes an entire multi-packet transfer. Larky '495, Col. 10, lines 13-31.

Claim 1, however, recites the limitation “the interfacing device including a configuration module for configuring, without the need of a CPU, the communication channel between the slave device and the host device.” Applicant respectfully submits that Larky does not teach this limitation. In fact, Larky requires the use of the device CPU to decode the “get descriptor” request and to set its memory pointer to the descriptor table 702 in RAM. Larky '495, Col. 10, lines 21-24. An illustration of the descriptor table is Table 1. Larky '495, Col. 10, lines 22-24. Table 1 is a list of generic endpoints and their types. The device CPU in Larky is required to set its memory pointer to the table, and then have its dedicated USB circuit complete a multi-packet transfer. Larky '495, Col. 10, lines 22-25. Apparently, according to Larky, a table like Table 1 is downloaded to the RAM in the device. Larky specifically states that, in accordance with his invention, a control transfer, such as the one shown in FIG. 4, is used to load the RAM 122 of the device controller with software code that contains the device's configuration information and endpoints are part of the device's configuration information. Larky '495, Col. 8, lines 8-13. Larky needs the device CPU to decode the “get descriptor” request and set the memory pointer to the descriptor table because the information is stored in the device's RAM 122.

In contrast, Applicant's invention has no such requirement. The configuration module configures the communication channel without needing a CPU for any of the steps. Thus, instead of reducing the number of CPU steps from seven down to two, as shown in FIG. 7b of Larky, Applicant's invention reduces the number of CPU steps to zero. Larky simply does not contemplate the possibility of reducing the CPU steps to zero during the configuration process, but instead maintains the use of the CPU during the process. Therefore, Larky '495 fails to teach each and every limitation of claim 1.

Regarding claim 2, the Office Action alleges that Larky teaches all the elements of the claim. Applicant submits that Larky does not teach the limitation “wherein the configuration module... includes a plurality of finite state machines that are operative to receive and respond to a request, including a GET_DESCRIPTOR and GET_CONFIGURATION request, from the host device.” In Larky, the device CPU decodes the “get descriptor” request. Larky, Col. 10, lines 21-22. In Applicant's invention, a state

machine receives and responds to the GET_DESCRIPTOR request, as recited in claim 2. Therefore, Larky fails to teach each and every limitation of claim 2.

Regarding claim 3, the Office Action has alleged that the argument of for claim 2 applies and that Larky teaches the additional limitations of claim 2. However, Applicant respectfully submits that Larky fails to teach the limitations of claim 3 at least because the reference fails to teach the limitations of claim 1 and that Larky does not teach the limitation “the second state machine having at least one output line for carrying a data transfer signal, and activating the data transfer signal based on the interpretation of the request packet.” Thus, in Applicant’s invention, the second state machine, in claim 3, is responsible for interpreting the request packet. In Larky, the request packet is interpreted (decoded) by the device CPU. See Applicant’s specification, page 13, lines 10-12. Therefore, Larky does not teach each and every limitation of claim 3.

Regarding claim 4, the Office Action again makes the same argument as was submitted for claim 3 and alleges that the additional limitations of claim 4 are taught as well. Applicant respectfully submits that Larky does not teach the limitations of claim 4 at least because it does not teach the limitations of claim 3. Additionally, claim 4 makes it clear that one of the requests is a GET_DESCRIPTOR request, and claim 3, from which it depends, makes clear that it is the second state machine that interprets this request, which is not taught by Larky. Therefore, Larky fails to teach each and every limitation of claim 4.

Regarding claim 7, the Office Action alleges that the argument for claim 1 applies and that Larky teaches the additional limitations of claim 7. Applicant respectfully submits that claim 7 is not taught by Larky at least because claim 1 is not taught by Larky. Additionally, in the cited portions of the reference, Larky is describing a state machine that contains endpoint configuration data and participates in the enumeration protocol. Larky, Col. 8, lines 28-31. Claim 7 is referring to the state machines associated with the functions of the function engine, which is the application-specific USB device. For example, a function engine is an A/D converter. While Larky mentions that other state machines perform USB related tasks under the control of the device CPU, it is unclear what these tasks are or whether they are the application-specific functions of the peripheral device. Larky, Col. 4, lines 32-35. Applicant believes that application-specific functions of the peripheral device are not tasks related to the operation of the Universal Serial Bus. Therefore, Larky fails to teach

each and every limitation of claim 7.

Regarding claim 8, the Office Action alleges that the argument for claim 7 applies and that Larky teaches the additional limitations of claim 8. Applicant respectfully submits that Larky fails to teach the limitations of claim 8 at least because it fails to teach the limitations of claim 7. Additionally, Larky fails to teach the limitation “wherein the type field is accessible by at least one of the state machines in the group associated with the endpoint.” One of the state machines in the group associated with the endpoint refers to the group of state machines associated with the function-engine endpoint, as recited in claim 7. As argued above, Larky fails to teach the use of state machines associated with a function-engine endpoint. Larky only teaches the use of state machines for USB-related tasks. Larky, Col. 4, lines 32-35. Therefore, Larky fails to teach each and every limitation of claim 8.

Regarding claim 10, the Office Action alleges that the argument for claim 7 applies and that Larky teaches the additional limitations of claim 10. Applicant respectfully submits that Larky fails to teach the limitations of claim 10 at least because it fails to teach the limitations of claim 7. Additionally, Larky fails to teach the limitation “wherein the group of state machines associated with the function engine endpoint includes: a data storage state machine that stores data sent from the host for the function engine; and a command state machine that interprets the stored data in the data storage machine to operate the function engine.” As discussed above, Larky fails to teach the use of state machines associated with the application-specific functions of the peripheral device. Larky only teaches the use of state machines for USB-related tasks. Larky has no mention of the use of a data storage state machine that stores data sent from the host *for the function engine*. Nor does Larky mention anything about a state machine that interprets stored data in the data storage machine *to operate* the function engine. Therefore, Larky fails to teach each and every limitation of claim 10.

Regarding claim 12, the Office Action alleges that the argument for claim 7 applies and that Larky teaches the additional limitations of claim 12. Applicant respectfully submits that Larky does not teach the limitations of claim 12 at least because the reference does not teach the limitations of claim 7. Additionally, Larky does not teach the limitations “wherein the group of state machines associated with the function engine endpoint includes: a data collecting state machine that holds data sent from the function engine for the host; and a

command state machine that interprets commands from the host to read the data in the data collecting state machine,” as explained with respect to claim 10.

Regarding claim 16, the Office Action alleges that the argument for claim 1 applies and that Larky teaches the additional limitations of claim 16. Applicant submits that Larky does not teach the limitations of claim 1 as explained above. Additionally, Larky does not teach the additional limitations of claim 16 because Larky does not teach the use of state machines for the application-specific function engine. Larky only teaches the use of USB-related state machines.

Regarding claim 17, the Office Action alleges that the argument for claim 16 applies and that Larky teaches the additional limitations of claim 17. Applicant respectfully submits that Larky does not teach the limitations of claim 17 at least because the reference does not teach the limitations of claim 16 and because Larky does not teach the limitation “wherein the group of state machines associated with the function engine endpoint includes a data transfer state machine which receives commands from the host for controlling the operations of the second endpoint associated with the function engine.” Larky teaches the use of state machines for USB related tasks, not the use of state machines for the application-specific function. Therefore, Larky does not teach each and every limitation of claim 17.

Regarding claim 18, the Office Action alleges that the argument for claim 1 applies and that Larky teaches the additional limitations of claim 18. Applicant respectfully submits that Larky does not teach the limitations of claim 18 at least because it does not teach the limitations of claim 1 and because Larky does not teach the limitations “a memory buffer having a plurality of storage locations and associated with the endpoint; and an endpoint register having an index field for addressing the memory buffer.” Larky has a memory buffer associated with the enumeration process on the USB, but never says how the application specific function operates, whether it has a memory buffer or whether the endpoint register has an index field for addressing the memory buffer.

Item 6

Regarding claim 5, Applicant has amended claim 5 to include the limitations of claims 3 and 1, from which it depended.

Regarding claim 6, Applicant has amended claim 6 to include the limitations of claims 3 and 1, from which it depended.

Regarding claim 9, Applicant has amended claim 9 to include the limitations of claims 7 and 1, from which it depended.

Regarding claim 13, Applicant has amended claim 13 to include the limitations of claims 12, 7 and 1 from which it depended.

Regarding claim 14, Applicant has amended claim 14 to include the limitations of claims 7 and 1 from which it depended.

Regarding claim 15, Applicant believes that claim 15 is allowable at least because claim 14, from which it depends, is allowable.

Applicant has added claims 20 and 21. Claim 20 adds the limitations to claim 1 of “a memory block for holding control information for the function engine; and a command state machine for interpreting and executing the control information to control a function of the function engine.”

Claim 21 adds the same limitations as claim 20 adds but to claim 7. Support for claims 20 and 21 can be found in the specification at page 9, lines 15-31 and FIG. 4.

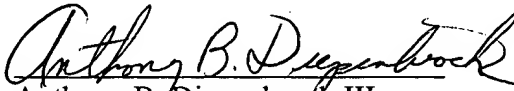
CONCLUSION

Thus, in light of the above, having responded to each and every ground of rejection, Applicants respectfully request reconsideration and allowance of the new and pending claims in the above-mentioned application.

Respectfully submitted,

DECHERT LLP

Dated: December 1, 2003

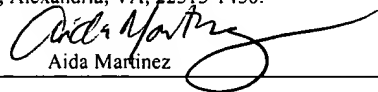

Anthony B. Diepenbroek III
Reg. No. 39,960

DECHERT LLP
Customer No. 37509
P.O. Box 10004
Palo Alto, CA 94303-0961
Tel: 650.813.4800
Fax: 650.813.4848

CERTIFICATE OF MAILING (37 CFR 1.10(a))

CERTIFICATE OF MAILING BY "EXPRESS MAIL" Rule 10: I hereby certify that this correspondence is being deposited on December 1, 2003, with the U.S. Postal Service "Express Mail Post Office to Addressee" under 37 CFR 1.10 as **Express Mail No. EL 989607791 US** addressed to: Mail Stop Fee Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA, 22313-1450.

Date: December 1, 2003


Aida Martinez